

A Land Use/Land Cover Change Geospatial CyberInfrastructure to Integrate Big Data and Temporal Topology

Jin Xing and Renee Sieber

Department of Geography, McGill University, Montreal, Canada

Email: jin.xing@mail.mcgill.ca; renee.sieber@mcgill.ca

Abstract

Big data has shifted spatial optimization from a purely computational-intensive problem to a data-intensive challenge. This is especially the case for spatio-temporal Land Use / Cover Change (LUCC) research. In addition to greater variety, for example from sensing platforms, big data offers datasets at higher spatial and temporal resolutions; these new offerings require new methods to optimize data handling and analysis. We propose a LUCC-based Geospatial CyberInfrastructure (GCI) that optimizes big data handling and analysis, in this case with raster data. The GCI provides three levels of optimization. First, we employ spatial optimization with graph-based image segmentation. Second, we propose ST Atom Model to temporally optimize the image segments for LUCC. At last, the first two domain spatio-temporal optimization is supported by the computational optimization for big data analysis. The evaluation is conducted using DMTI (DMTI Spatial Inc.) Satellite Streetview imagery datasets acquired for the Greater Montreal area, Canada in 2006, 2009, and 2012 (534 GB, 60cm spatial resolution, RGB image). Our LUCC-based GCI builds an optimizing bridge among LUCC, spatio-temporal modelling, and big data.

Keywords: LUCC; Geospatial CyberInfrastructure; Optimization; Spatio-Temporal Object Model

1. Introduction

Geographic Information Science (GIScience) and Remote Sensing (RS) research into big data has been triggered by increasing spatial, spectral, and temporal resolutions of sensing systems and Web 2.0 platforms (McAfee et al., 2012). That is, we simply have magnitudes' larger volumes of data, which are arriving at increasing velocity, and with

greater variety in data structures. Big data has forced a rethinking of numerous aspects of GIScience, from spatial data collection and storage to sampling, analysis, and visualization (Hampton et al., 2013; Liang et al. 2010; Zaslavsky et al. 2013). Big data also requires a new architecture for managing those methods.

Big data has the potential to shift research on detection of Land Use/Cover Changes (LUCC). The field of LUCC has been explored for over 50 years (Singh, 1989). LUCC detection addresses three questions: (1) Is there any change of interest when comparing two or more temporally distanced datasets?; (2) What are these changes quantitatively?; and (3) What are the change trajectories and corresponding rates? Because big data affects spatial and temporal domains simultaneously, it impacts all three questions. Higher volumes and velocity may allow us to detect finer grained changes that may have been missed with datasets at coarser spatial resolution and temporal periodicity. Techniques in quantitative detection of LUCC should enable multi-temporal analysis (comparison of more than two raster datasets) and handle heterogeneity in spatial, spectral and temporal resolutions. Big data dramatically increases the number of potential changed objects (since more objects can be extracted from higher spatial and spectral resolutions and more object changes can be detected with improved temporal resolutions). Big data promises greater LUCC but renders the trajectories of those changes—time—more difficult to delineate.

Additionally, big data poses significant computational challenges, such as the need for scalable data storage, flexible computing resource provisioning, and dynamic workflow management. Solutions to these challenges should be integrated with the domain demands of LUCC to increase accuracy of results and shorten computation time.

In this paper, we propose a LUCC-based Geospatial CyberInfrastructure (GCI) that seeks to optimize the geospatial handling and analysis of big data. This optimization is three-fold. A domain-based layer handles spatial optimization through energy cost minimizations of pixel clustering into feature objects. Because LUCC in big data likely requires handling multiple time slices, our GCI temporally optimizes via what we call a Spatio-Temporal (ST) Atom Model. Third, our GCI optimizes computing resource provisioning, data decomposition, and workflow. Figure 1 shows how these optimizations function in our GCI.

[Figure 1 here]

The paper is organized as follows. In Section 2, we discuss research to date on LUCC and spatio-temporal modelling. We describe our LUCC-based GCI, which attempts to optimize along space, time, and data handling in Section 3. We deploy and evaluate the optimization methods in Section 4. We conclude with opportunities for future research.

2. LUCC and ST Optimization

In this section, we review the related works about employing spatial optimization in LUCC. Then we delineate the needs of using temporal information to optimize an object-based LUCC. We also discuss literature on computational optimization within GCI for LUCC support.

2.1 Optimization Challenges in LUCC

Spatial optimization has been studied in LUCC for a long time (Tong and Murray, 2012). Jenerette and Wu (2001) utilize spatial optimization to simulate the LUCC in the central Arizona–Phoenix region of U.S., while Ligmann - Zielinska et al. (2008) optimize generative models for land-use allocation. Most of these works focus on spatial optimization at a given time, but temporal information has rarely been integrated

into the optimization process.

A traditional approach in LUCC for identifying change has been pixel-based. The pixel-based approach relies on pixel-level calculation to generate a “difference image” (e.g., the subtraction of two images) to identify the relative amounts of change. This approach is frequently utilized for bi-temporal analysis, and generally requires imagery datasets to match in spatial and spectral resolutions (Singh, 1989). Reviews of LUCC (Singh, 1989; Lu et al., 2004; Jianya et al., 2008) have revealed a gradual shift in research from pixel-based to object-based approaches, which work with groups of pixels as objects. In part, this is because increasing spatial resolutions afforded by big data allow for pixels that are significantly smaller than objects of interest. Object-based approaches have the additional advantage of moving us beyond traditional raster-vector divides that separate RS from GIScience.

Blaschke (2010) terms the shift to objects as OBIA. OBIA groups similar image pixels as objects, calculates object features, and then applies classification algorithms (Congalton 1991) to label various types of changes. Walter (2004) argues that OBIA is less sensitive than pixel-based analyses to different spatial and spectral resolutions of datasets because comparing object properties (e.g., texture and shape) can identify change. According to Tong and Murray (2012), OBIA is a type of district optimization, which spatially optimize the change areas for multi-temporal LUCC. Also, because pixels are grouped into and then conceptually handled as objects via spatial optimization (Batz and Schäpe, 2000), OBIA attempts to represent some degree of spatial topology in each original image with these objects.

OBIA has its drawbacks. First, if parts of the object change from time t_1 to t_2 then OBIA may mark the entire resultant object as “changed”. Figure 2 illustrates this issue. Second, the temporal topology is not retained in the OBIA process. The lack of

temporal topology change information impedes integration of OBIA into multi-temporal LUCC, which tracks the impact of one object's change on its neighbours over several periods (Pijanowski et al., 2002). If we choose OBIA then we should find techniques that optimize the integration of change information in both spatial and temporal dimensions, including the between-time topology.

[Figure 2 here]

Figure 2 illustrates the problems in OBIA with two images of Montreal, Canada, at 2009 (t_1) and 2012 (t_2). We show sample objects generated from clustering: at t_1 , a forest object and at t_2 , a forest-“donut hole” object and a building object. The building object compared to the spatial extent of the forest object t_1 results in the whole object being labelled as “changed”. From Figure 2, we can see only using spatial optimization for image segmentation cannot solve the partial object challenge, so we need to use the temporal information to further optimize the OBIA.

Before we arrive at objects, we need to group pixels according to their similarities. This is called image segmentation and follows three general types: spatial segmentation, feature-based clustering, and graph-based methods. Spatial segmentation extracts regional entities from imagery datasets based on the spatial structure information, while feature-based clustering algorithms rely on similarities of image features to group pixels. Graph-based image segmentation method combines elements of spatial- and feature- based image segmentation methods (Shi and Malik, 2000). The core idea of the graph based method lies in constructing a weighted graph, where each vertex represents pixels (regions) in the image and the weight of each edge connecting two pixels represents the likelihood of segmentation. The weight, which is usually calculated by combining feature and spatial information, forms a cost energy function. Minimization of the energy cost is considered a traditional spatial optimization, where

the image is cut into several segments (Tong and Murray, 2012). Graph based segmentation algorithms have been studied extensively for object extraction (Sumengen and Manjunath, 2006; Jermyn and Ishikawa, 2001; Wu and Leahy, 1993).

The spatial optimization found in energy cost minimization suffers from difficulties in determining initial values and is easy to trap with the local optimal solutions (Celik and Yetgin, 2011). Various parametric learning and optimization approaches have been applied with graph based segmentation methods to address these problems. For example, parametric maxflow (Gallo et al., 1989) integrates non-local feature into the optimization process; Kolmogorov et al. (2007) present case studies using maxflow approach. Lempitsky et al. (2012) propose a global optimization method called “branch-and-mincut” for graph based image segmentation with segmentation mask and non-local parameters. To handle multi-label image segmentation, Boykov and Funka-Lea (2006) supply α -expansion and $\alpha\beta$ -swap-move based algorithms.

Building the graph and the minimization of the energy cost function are highly computation-intensive, and may take a long processing time. Since most research projects have time constraints, it is very tempting to consider High Performance Computing (HPC). With big data, this process becomes both data-intensive and computationally-intensive, so it becomes more difficult to consider this optimization process separately from the computation. We conclude that optimization in big data LUCC requires a combination of spatial, temporal, and computational optimization methods.

2.2 ST Modelling and Temporal Optimization in LUCC

Different ST models have been integrated with LUCC (Radke et al., 2005). These include statistical distribution modelling of the change and non-change areas (Bazi et al., 2005), predictive models (Veldkamp and Lambin, 2001), and cellular automata

simulations (Li and Yeh, 2002). These applications advance temporal and spatial components, but not the two components equally (Deng et al., 2009; Pan et al., 1999).

Over the years, GIScience researchers have proposed various methods to effectively and elegantly integrate temporal and spatial dynamics. Yuan (1996) provided the first survey of ST models, which illustrated their pros and cons in representing LUCC. More recently, Nandal (2013) reviewed ST models, and categorized them into ten types: snapshot model (Armstrong 1988), space-time composite data model (Langran and Chrisman, 1988), data models based on simple time-stamping (Allen, 1991), event-oriented model (Peuquet and Duan, 1995), three domain model (Yuan, 1999), history graph model (Van Der Wal and Pye, 2003), Spatio-Temporal Entity-Relationship (STER) model (Parent et al., 1999), Object-Relationship (O-R) model (Coppin et al., 2004), ST object model (Huang and Chandramouli, 2009), and moving object data model (Erwig et al., 1999).

A predominant reason why so many models have been created is that it is difficult to determine how best to generate and store the changes. For example, missing state information creates difficulty in applying event or process based temporal modelling (i.e., event-oriented model, O-R model, STER Model, and moving object data model). A challenge in applying, for example, the space-time composites model occurs when attempting to compare imagery datasets with heterogeneous resolutions, which prevents the direct overlay of temporal snapshots of land use (Nadi and Delavar, 2003). Likewise, the difficulty of extracting semantics from imagery datasets impedes the employment of the three domain model. Most ST models are vector based, at least in their deployment; whereas, RS imagery datasets are generally raster data. ST models need to provide interfaces to ease the vectorization process. The simple time stamping and the history graph model method present difficulties in vectorizing RS datasets. ST

models, we argue, pose a significant optimization problem that will only get worse with big data.

2.3 GCI Related to LUCC

GCI has been designed to handle challenges found in big data research and in computation intensive jobs found in GIScience and RS (Wang, 2010). For example, Liang et al. (2010) used GCI to enable sharing and visualization of big environmental sensing datasets. Yue et al. (2010) proposed a semantic web based GCI to provide on-demand RS big data products. A GCI was also developed to perform data mining from volunteered geographic information harvested over the Internet (Gao et al., 2014).

GCI can provide the integration of domain specific optimization and computing techniques for GIScience. Consequently GCI research is intertwined with specific hardware and software platforms for distributed data handling. An example of this intertwining is Xia et al.'s (2010) hardware solution—a CUDA (Compute Unified Device Architecture) based GCI to accelerate inverse distance weighting and viewshed analysis. CUDA exacts a cost in host-device data transfer, which cannot be neglected in large volume transfers (Yang et al., 2008).

Compared to hardware (e.g., CUDA, grid computing), Yang et al. (2011) conclude that cloud computing affords the best platform for geospatial big data. Specific cloud solutions include Google's development of MapReduce, which is a software platform to distribute computing tasks over multiple machines. Hadoop, an open source implementation of MapReduce, is highlighted by Yang et al. (2010) for its capacity to process big spatial data.

Hadoop is the preferred choice for GCIs due to its scalability and flexibility (Nurian et al., 2012). Nonetheless, it has problems. Lee et al. (2012) highlight the weakness of dataflow management in MapReduce. They also note the low input-output

efficiency of MapReduce. Some researchers have begun to explore data streaming, which is defined as a continuous sequence of datasets. Researchers have implemented data streaming to analyze radar datasets (Plale 2006). Another study utilized data streaming for environmental observation analysis in cluster computing (Tilak 2007). Neither study calls their work GCI; however, they resemble GCIs in that geospatial analysis is conducted with distributed computing environment and the emphasis is on the underlying architecture.

GCIs have not been widely applied for big data analysis in LUCC. First, domain specific optimization challenges in LUCC require scalable and flexible computing resource provisioning. Second, the massive data exchange among different computation process which shape LUCC are much more complicated than a collection of batch processing. Third, a LUCC workflow also needs to be optimized for better data transfer and less computation time. Therefore, the optimization of computing resource provisioning and workflow management need to be twisted together with LUCC studies. In this paper, we propose LUCC-based GCI, to provide the integrated GCI-based optimization.

3. LUCC-based GCI

[Figure 3 here]

Figure 3 illustrates the architecture of our LUCC-based GCI. This GCI provides the integration of domain specific optimization methods with computation optimization techniques. Specifically, our spatial and temporal optimization methods extract what we call ST atoms from multi-temporal images to detect any changes, where the ST atoms stand for image pixel groups that either remain or completely change across the time span. The whole process is supported by data streaming, Voronoi image decomposition, workflow optimization and scalable cloud computing resources. In the era of big data,

we argue that we should consider optimization as a combination of domain knowledge and computation (Wang, 2010). Otherwise, excessively long processing times incurred, for example by “oversplitting” of big data due to insufficient understanding of a domain like LUCC, can hinder the knowledge discovery in big data.

3.1 Optimization in the Domain Layer

[Figure 4 here]

Figure 4 shows the workflow of our domain layer, focusing on our optimization methods. First, we implement a graph-based image segmentation process to extract objects with spatial and spectral similarities from multi-temporal RS imagery datasets. Second, we use temporal topology rules to find the ST atoms with the image segments. Finally, we generate the change trajectories based on the ST Atom Models by applying classification and ST object modelling.

3.1.1 Graph Based Image Segmentation with Spatial Optimization

We use $\mathbf{X}_{t_l} = \{x_{t_l,1}, x_{t_l,2}, \dots, x_{t_l,I}\}$ to denote an image that is recorded in time t_l with $I \times J$ pixels and b bands. \mathbf{X}_{t_l} is modelled as an undirected graph $\mathbf{G}: (\mathbf{V}, \mathbf{E})$, where the pixel in spatial position (i,j) is linked with a vertex $v_{i,j} \in \mathbf{V}$, and $e_{i,j;t,u} \in \mathbf{E}$ is the edge that connects $v_{i,j}$ to its neighboring pixel $v_{t,u}$. In this paper, we consider the neighbouring system N as a 4 connected grid, which consists of ordered pixel pairs $(x_{i,j}, x_{t,u})$. We introduce $\mathbf{L} = \{1, 2, \dots, \mathbf{K}\}$ and \mathbf{K} labels. \mathbf{K} labels are defined for the given image, for multi-object segmentation. Let $f = \{f_{v_{i,j}} | v_{i,j} \in \mathbf{X}_{t_l}\}$ ($f_{v_{i,j}} \in \mathbf{L}$) be the collection of all the pixel-label assignment. The spatial optimization is found via an energy cost function. The energy function of our graph based image segmentation is formulated, according to Boykov et al. (2001), as:

$$E(f) = \lambda \sum_{v_{i,j} \in \mathbf{V}} D(f_{v_{i,j}}) + \sum_{v_{i,j}, v_{t,u} \in \mathbf{N}} V(f_{v_{i,j}}, f_{t,u}) \quad (1)$$

The term $D(f_{v_{i,j}})$ is called the data term and $V(f_{v_{i,j}}, f_{v_{t,u}})$ is named the smoothness term. $D(f_{v_{i,j}})$ represents the cost of assigning label $f_{v_{i,j}}$ to pixel $v_{i,j}$; whereas $V(f_{v_{i,j}}, f_{v_{t,u}})$ penalizes spatial inconsistency and tends to assign the same label to neighbouring pixels. Minimizing $E(f)$ will optimize segmentation in the image graph. A solution to the multi-labelling problem is achieved by the α -expansion algorithm (Boykov et al., 2001). Given a labelling f and a label α , a move from f to f^α is called an α -expansion if $f_{v_{i,j}} \neq f_{v_{i,j}}^\alpha \rightarrow f_{v_{i,j}}^\alpha = \alpha$. The α -expansion algorithm iterates over all labels α to find the best α -expansion until convergence. The drawback is that α -expansion might trap the local optima. The trapping problem means that a local optima (either maximum or minimum) within a neighbouring set of candidate solutions is mistakenly considered as the global optima of all the candidates. To overcome this problem, we utilize the global optimization branch-and-mincut algorithm (Lempitsky et al., 2012).

To achieve the minimal value in the energy cost function, we use the branch-and-mincut spatial optimization method. This method tends to find the global optimal solution by a top-down propagation of the active node in the feature space, which is organized as a binary tree. This technique is built on top of graph cut and branch-and-bound algorithm (Quesada and Grossmann, 1992). Lempitsky et al. (2012) has proved its effectiveness in different image segmentation studies.

We turn to the data term and smoothness term in Eq (1). It is quite difficult to calculate the data term $D(f_{v_{i,j}}) = -\log \Pr(f_{v_{i,j}}|F_{v_{i,j}})$ directly, where $F_{v_{i,j}}$ is the observed geometric feature vector for pixel $v_{i,j}$. Liu et al. (2008) propose a super-pixel and SVM (Supporter Vector Machine) classification-based method to approximate the distribution of $\Pr(f_{v_{i,j}}|F_{v_{i,j}})$. We adopt Liu et al.'s (*ibid.*) approach in this paper.

$V(f_x, f_y)$ reflects the weight among pixels in the graph. The weight exists to penalize assigning different labels to adjacent pixels. The RBF (Radial Basis Function) kernel (Camps-Valls et al., 2008) is used for its (relative) simplicity:

$$\omega_{i,j;t,u} = e^{\frac{-\|I_{i,j}-I_{t,u}\|_2^2}{2\sigma^2 \text{dist}(I_{i,j}, I_{t,u})}} \quad (2)$$

where $I_{i,j}$ is the intensity of $x_{i,j}$, $\text{dist}(I_{i,j}, I_{t,u})$ stands for the spatial distance, and σ is the Gaussian width. We illustrate image segmentation using RS image $\mathbf{X}_{t/l}$. The equations will be similar when comparing multiple images. Using the graph based optimization method, we can obtain a collection of image segmentations for multi-temporal RS imagery datasets.

3.1.2 Spatio-Temporal Atom Extraction and Labelling

We now need to extract ST atoms using temporal optimization, which guarantees the ST atoms either remain the same or completely change through the study time span. The ST Atom Model is proposed to handle the partial object changes in Figure 2, and amend the lack of temporal topology in LUCC. To implement the ST atoms in LUCC we use the largest homogeneous units that hold their spatial and temporal features. We find that the object concept in ST model and OBIA can most easily be applied to the ST atoms. ST atoms can be viewed as hybrid vector-objects, which become the lingua franca between OBIA and the ST object model in LUCC.

For pre-processing, we need to first “snap” the image segments into objects. The image segmentation boundary pixels obtained in Section 3.1.1 might be discrete and out of order. For example, the boundary of image segments may not form neat lines and may contain numerous small variations. To facilitate the employment of ST atom extraction, we utilize a chain-code (Li et al., 1995) to connect the boundary pixels and then we

implement the Douglas-Peucker algorithm to turn the region boundary into polygons (Saalfeld, 1999). These processing steps turn the image segments into objects. Then object matching is conducted using coordinates of image registration to find the corresponding objects in the other time periods.

[Table 1 here]

The most important part of our temporal optimization is retaining the temporal topology. The temporal optimization process can be defined as finding the largest ST atoms with the constraints of temporal topology rules. Egenhofer and Al-Taha (1992) and Müller and Zeshen (1992) provide a set of spatial rules that explain spatial relations among objects (Table 1). Concepts of “equal”, “contain”, “split”, “overlap”, “merge”, and “covered-by” rules (Egenhofer and Al-Taha, 1992) are useful to model the interactions between two ST atoms with similar spatial extents. [Other topology change rules, like, “disjoint” in Egenhofer and Al-Taha (1992) are not included as they have little impact on the ST atom extraction.] In addition to considering the spatial distance between objects (Egenhofer and Al-Taha, 1992), we highlight the temporal topological changes from t_1 to t_2 . We introduce the state of “partial change” to describe situations in which Object A cannot be considered having completely changed into one or more objects. An example of this would be parts of a roof on a hypothetical Building A that are re-painted. The “expand” and “shrink” topology change rules describe changes that involve all neighboring objects (Müller and Zeshen, 1992). The key idea is to keep ST atoms as the largest temporally homogeneous object. An object is split into ST atoms, which will be entirely changed or unchanged across the LUCC study time span. There is no partially changed ST atom after the employment of these rules.

Nine rules for bi-temporal topology change are listed. More complicated topologies can be represented by combining two or more of these nine basic rules. For example, we can obtain three objects from Figure 2 t_l by simply applying the “contain” rule twice with a bi-temporal image pair. The ST atom extraction also can be parallelized to fit the distributed computing environment in our LUCC-based GCI. The ST atom extraction process thus becomes the temporal optimization process for OBIA.

We first implement change masks. Change masks serve as templates for any interpolation, for results of temporal rule application, and for difference images that will then be overlaid onto the original objects to extract the atoms. Image interpolation is applied if the two candidate objects do not have the same spatial resolution (Lam, 1983). We perform interpolation at this stage, instead of at the pre-processing stage, because the graph based image segmentation and classification parameters are very sensitive to the noise generated by overall image interpolation (Lempitsky et al., 2012). Then we utilize the univariate image differencing technique (Singh, 1989) to generate the “difference image”. By employing *k-means* clustering algorithms (Rui and Turi, 1999) and a thresholding technique (Lu et al., 2004), the difference image is clustered as several change areas. Temporal topology rules are applied to each change mask, which are overlaid onto the original objects to generate ST atoms. Atom extraction processes are applied in an iterative bi-temporal way. Each time we apply one entry in Table 1 to extract ST atoms with the bi-temporal image pairs. An iterative application of temporal topology rules transforms the objects into ST atoms (more details in Section 4).

The final process employs a classification algorithm, where the ST atoms are given labels to represent the actual LUCC types (e.g., forest, buildings, roads, and grassland). Because we use OBIA, the label is not ascribed to individual pixels but to ST atoms. A broad range of classification algorithms can be used to generate the labels

(Walter, 2004). We use the SVM classification algorithm due to its high classification accuracy and low sensitivity to noisy data in RS analysis (Melgani and Bruzzone, 2004).

We create a ST LUCC optimization that can be widely applied with RS imagery datasets that possess a high level of variety. The increased variety of big data suggests that LUCC must be adjusted according to imagery datasets with heterogeneous spectral and spatial resolution. To some extent, our OBIA with optimization technique eliminates spatial and spectral heterogeneity (Chen et al., 2010); whereas temporal topology rules addresses the temporal heterogeneity. We also note if the spatial and spectral resolutions are very different (e.g., one with 0.6m spatial resolution and the other one with 600m), our LUCC optimization may fail.

3.2 HPC and Workflow Management Optimization

Workflow management is a large portion of any GCI. The LUCC-based GCI dataflow management layer partitions the big datasets as shown in Figure 3. We use the Voronoi diagram and Fortune's Sweepline algorithm as described in Xing and Sieber (2014) to better decompose large datasets. This method provides rough load balancing and minimizes the influence of the splitting borders in LUCC. It serves as a way to optimize splitting tiles so that features are retained as much as possible.

Our Voronoi-based partitioning method uses data streams afforded by Apache Storm, which also serves the HPC. Storm is a free and open source software project of the Apache Software Foundation. Storm relies on data streaming as part of real-time job scheduling to improve parallel computing support for big data analysis. Data streams can be fed into any number of processing nodes with minimum couplings in parallel. Storm characterizes streams as an unbounded sequence of data tuples. Data tuples may contain image tiles, object or ST atoms. Tuples are continuously pushed into processing

nodes in parallel. Storm manages the network of these data streaming communications, which is called *topology* management. The Storm framework is the development tool to build the data stream topology and provide fault-tolerance functionalities in our LUCC-based GCI.

The optimization of LUCC detection computation can be formulated as the minimization of computation time with respect to limited computing resource. Graph image segmentation and ST atom extraction illustrate how Storm, the software, and data streaming, the concept, achieve computational optimization. Given equal computing resources (i.e., virtual machines {VMs} with identical configuration), it is unlikely that all nodes running the image segmentation tasks will finish at the same time. There will be a moment when some nodes finish their tasks while others are still running. Our LUCC-based GCI streams image segments to ST atoms extraction VM, while other image segmentation jobs are still running.

[Figure 5 here]

We choose Storm as opposed to Hadoop. Compared with a Hadoop based GCI (Li et al., 2011), we argue that Storm in our GCI offers better flexibility and efficiency in job scheduling and dataflow management. Figure 5 shows a comparison between Storm and Hadoop. A Hadoop based GCI must wait until all VMs finish their image segmentation tasks. ST atom extraction can be scheduled in parallel with image segmentation on Storm. Another advantage of a Storm based GCI is the use of data streams instead of intermediate data storage (e.g., HDFS {Hadoop Distributed File System} in Hadoop) for image segmentation results. This reduces input-output costs. At last, the cloud computing is employed to provide the scalable and flexible computing resource provisioning.

4. Results

We designed two case studies to test our LUCC-based GCI. The first case tested the performance of our LUCC-based GCI to optimize the LUCC detection. The second case compared two different spatial optimization algorithms in multi-temporal LUCC detection.

Figure 6 and Table 2 show details of the implementation. The test bed was deployed on the Amazon Elastic Cloud Computing platform (Amazon EC2, 2014). Seventy-one VMs were utilized, with one local GCI controller and 70 nodes in the EC2 cloud. Sixty data streams were configured, with 30 streams connecting graph image segmentation VMs to ST atom extraction VMs. Twenty were utilized for the iterative ST atom extraction communication. The ST atom extraction VMs were connected to 10 classification and ST modelling VMs via 10 streams.

We used the Storm framework to map connections between VMs. During the implementation on Amazon, we utilized Amazon Kinesis to perform the actual data streaming. Kinesis is a data streaming service to manage real-time communication among Amazon cloud computing components (e.g., VMs) (Amazon Kinesis, 2014). Kinesis replaces the default stream of Storm and has been found to exert no impact on Storm's functionality (Bhartia, 2014). Then we utilized the topology interface of Storm to specify the topology of computing nodes and data streams.

[Figure 6 here]

The evaluation imagery datasets were collected in 2006, 2009, and 2012 for the Greater Montreal area, Canada (DMTI, 2006, 2009, 2012), with total size of approximately 534GB. The DMTI StreetView RGB satellite images are recorded at 0.6m spatial resolution; we assume that no object is smaller than one pixel. We employ radiometric correction, geometric rectification, image registration to each of the testing

datasets, and Voronoi image decomposition on the local controller node. Figure 6 shows the three steps of our ST Atom Modelling are hosted on Amazon EC2 cloud computing.

[Table 2 here]

4.1 ST Optimization

We first tested our ST Atom Modelling in the LUCC-based GCI. After pre-processing, image tiles (i.e., Voronoi polygons) were uploaded onto Amazon cloud. A week was required to upload image tiles onto Amazon S3 cloud storage (Amazon S3, 2014), which highlights the input-output challenges in big data. The second column of Table 3 lists the computation time of each step in our LUCC-based GCI.

We first needed to parameterize the graph cut. We used a subset of the tiles (i.e., 5 image tiles sampled from different areas of Montreal) to extract parameters for the cost energy function. Parameters were applied to the whole dataset for image segmentation. To avoid over-fitting, a cross validation technique was applied (Hall and Koch, 1992). One hundred ground truth points were selected from other image tiles to fine-tune the parameters. Then graph cut image segmentation was executed with the branch-and-mincut optimization. Finally, we followed Blaschke et al. (2000) and filtered objects unrelated to LUCC (e.g., vehicles).

The graph based image segmentation method generated 42,628 objects in 2006, 49,894 objects in 2009, and 47,742 objects in 2012. This represents relatively small differences (+17% and -4%) and reflects a depressed retail and residential market relative to other North American cities (e.g., CMHC, 2012). The Greater Montreal area has regions of dense urbanism but also is composed of agriculture and forest. We found a large number of objects located in urban portions of our study area, due to a high mixture of land uses.

ST atom extraction began with pre-processing, which included spatial interpolation, chain-code, and Douglas-Peucker algorithm. We did not employ spatial interpolation since all datasets matched in spatial and spectral resolutions. We found some instances of non-contiguous potential atoms so we utilized chain code together with a regression technique (Esbensen et al., 1992) and Douglas-Peucker to “snap” potential atoms in objects.

Temporal topology rules were applied to the change masks, which were then applied to extract the ST atoms from objects. The most applied temporal topology rules were “overlap” and “split”, which resulted in the large number of ST atoms. For example, the forest area in Figure 7 for 2006 was split as finer ST atoms, which either changed into buildings or remained forest across the whole study time span. Because there is some overlapping of the temporal topology rules (i.e., an object change can be classified as both "contain" and "partial change") and occasionally multiple rules can apply to one ST atom extraction. Building 4 in Figure 7 could utilize both “split” and “contain” rules. We chose the “split” rule because we utilized the 2006 objects as the “Object A” in Table 1. When multiple temporal topology rules are applicable, it is the implementation of the rules that determines the order of the rules’ application. So care must be taken in coding. The bi-temporal ST atom extraction output 228,683 ST atoms. The tri-temporal process, which compared 2006-2009 and 2009-2012, generated 750,402 ST atoms.

Finally, we used the SVM classification algorithm to label seven different types of atoms (i.e., forest, grass, farmland, bare ground, water, roads and buildings). The SVM training process was conducted according to Melgani and Bruzzone (2004) with ST atom features, like brightness, shape, and texture. SVM classification and ST modelling labelled 750,402 ST atoms with the pre-defined seven classes, time-stamp the

classified atoms, and linked them in time sequential order as ST Atom Models. “Forest” occupied the largest areas in greater Montreal; whereas “building” class dominated in number.

One thousand ground truth sampling points were randomly chosen to evaluate the performance of our ST optimization. These 1000 points were visually inspected and assigned labels. The highest accuracy was achieved in forest areas, approximately 97 percent. The lowest accuracy is found in complex urban areas, approximately 85 percent. The reason for this reduced accuracy is that ST atom extraction generates excessive numbers of atoms when there are complex object mixtures and numerous iterations of temporal topology rules. This results in poorer classification performance. For example, temporal topology rules may split one road into small blocks due to road repair in one study time period. Small roadblocks may possess very similar geometric and texture attributes and may be misclassified as buildings.

Figure 7 shows an example of new buildings that were constructed in a forest in 2009. Our ST optimization was able to detect these changes and presented them as ST atom models. Our ST Atom Model assures no partial ST atom changes across the study period (the “forest” ST atoms either remain as “forest” {ST Atom1 and 2} or completely into “buildings” {ST Atom 3 and 4}), because we optimize change information in both spatial and temporal domains. This evaluation proves the LUCC-based GCI can address partial object changes using ST Atom Modelling for big data analysis.

[Figure 7 here]

4.2 Spatial Optimization Comparison

Image segmentation optimization plays a pivotal role in our GCI-based optimization framework. We assessed its effect by comparing the min-cut/max-flow and branch-and-

mincut algorithm. Another 1000 ground truth points were selected to evaluate two instances of LUCC from 2006 to 2009, and 2009 to 2012 in greater Montreal area, Canada. We used a simple atom change/no-change error matrix (Macleod and Congalton, 1998) with reference data to test the performance of the two optimization algorithms. We used average accuracy from the change of seven predefined classes, and merged them as “change” and “no-change” super-classes. Overall accuracy can be calculated by adding the true change (change/change) and true no-change (no-change/no-change) percentage in Table 4. For 2006-2009 LUCCdetection, min-cut/max-flow achieved 97.2 percent in overall accuracy; whereas branch-and-mincut was 98.0 percent. Min-cut/max-flow optimization produced 96.6 percent in 2009-2012 LUCC detection, but branch-and-mincut slightly outperformed with 97.3 percent. Both algorithms generated satisfying results, but the performance of branch-and-mincut was slightly better than min-cut/max-flow. The reason could be the best-first branch-and-bound search mechanism of branch-and-mincut, which determines the global optima with searching tree techniques (Lempitsky et al., 2012). Nonetheless, this result does not guarantee branch-and-mincut will always outperform the min-cut/max-flow algorithm in graph based image segmentation. Further study is needed to find a suitable global optimization technique for LUCC detection.

[Table 4 here]

To evaluate our temporal optimization, we use the same 1000 ground truth points to compare the performance between the ST Atom Model and a standard OBIA change detection method (Chen et al. 2010). The overall accuracy of OBIA was 77.4 percent, which was 19.9 percent less than our ST Atom Model. We visually inspected the points and found OBIA mismarked 146 unchanged points as “change”, which were caused by

the partial object changes (see Figure 2). Additionally, the standard OBIA output temporally isolated objects, which prevented the generation of change trajectories.

Finally, we tested the computing optimization induced by Storm with a Hadoop version. Results are shown in Table 3. Hadoop required 168.7 hours, 29.6 percent more than the streaming implementation. Extra time was induced by HDFS based data exchange and unnecessary waiting time (Figure 5). Delays from the previous step accumulated in later steps, which explained the increasing delay in ST atom extraction, and classification and modelling steps in the Hadoop implementation.

5. Conclusion

In this paper, we presented and evaluated a GCI-based ST optimization for LUCC. Optimization techniques play important roles in the GCI, including the spatial, temporal and computational optimization techniques. With GIScience become data-driven (Miller and Goodchild, 2014), GCI shifts as an important knowledge discovery approach. Thus we can say the boundary between domain optimization and computation optimization will disappear in the future research.

With the ever-increasing amounts and speed of data, GCIs should integrate new methods for improving input-output and computing task scheduling. Despite new algorithms, more prosaic optimization of data handling is likely to constrain usage of GCIs. Advanced optimization algorithms also should be explored to improve the accuracy of image segmentation and LUCC detection. For example, algebraic geometry optimization (Wang, 2014) can improve image segmentation, and swarm optimization can optimize change/unchange thresholds (Liu et al., 2014). This paper provides a preliminary step in re-shaping optimization as a combination of domain knowledge and computation. More work can be done to optimize ST models for LUCC. The ST model cannot express change rates explicitly and has limitations in describing changes

semantically. This challenge might be solved by using Yuan's (1999) three domain model to include additional semantic information in the form of description tags for changes. On the other hand, the relationship between domain specific optimization and different computation optimization techniques also calls for further exploration. Hopefully, new methodologies of ST optimization within GCI will remain a focus in future research.

References

- Aggarwal, C. C. (2007). Data streams: models and algorithms (Vol. 31). Springer.
- Allen, J. F. (1991). Time and time again: The many ways to represent time. *International Journal of Intelligent Systems*, 6(4), 341-355.
- Amazon Elastic Cloud Computing (EC2) (2014). Online: <http://aws.amazon.com/ec2/>
- Amazon Kinesis (2014). Online: <http://aws.amazon.com/kinesis/>
- Amazon Simple Storage Service (S3) (2014). Online: <http://aws.amazon.com/s3/>
- Apache, Apache Storm (2014). Online: <https://storm.incubator.apache.org/>
- Armstrong, M. P., (1988). Temporality in spatial databases. *Proceedings: GIS/LIS'88*, 2:880-889.
- Baatz, M., & Schäpe, A. (2000). Multiresolution segmentation: an optimization approach for high quality multi-scale image segmentation. *Angewandte Geographische Informationsverarbeitung XII*, 12-23.
- Bazi, Y., Bruzzone, L., & Melgani, F. (2005). An unsupervised approach based on the generalized Gaussian model to automatic change detection in multitemporal SAR images. *Geoscience and Remote Sensing, IEEE Transactions on*, 43(4), 874-887.
- Bhartia, R., (2014). Implement a Real-time, Sliding-Window Application Using Amazon Kinesis and Apache Storm. (2014). Retrieved December 4, 2014, from <http://blogs.aws.amazon.com/bigdata/post/Tx36LYSCY2R0A9B/Implement-a-Real-time-Sliding-Window-Application-Using-Amazon-Kinesis-and-Apache>
- Blaschke, T. (2010). Object based image analysis for remote sensing. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(1), 2-16.
- Blaschke, T., Lang, S., Lorup, E., Strobl, J., & Zeil, P. (2000). Object-oriented image processing in an integrated GIS/remote sensing environment and perspectives for environmental applications. *Environmental information for planning, politics and the public*, 2, 555-570.
- Boykov, Y., & Funka-Lea, G. (2006). Graph cuts and efficient ND image segmentation. *International Journal of Computer Vision*, 70(2), 109-131.
- Boykov, Y., Veksler, O., & Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(11), 1222-1239.

- Camps-Valls, G., Gómez-Chova, L., Muñoz-Marí, J., Rojo-Álvarez, J. L., & Martínez-Ramón, M. (2008). Kernel-based framework for multitemporal and multisource remote sensing data classification and change detection. *Geoscience and Remote Sensing, IEEE Transactions on*, 46(6), 1822-1835.
- Canada Mortgage and Housing Corporation, (2014). Housing Market outlook, Montréal CMA. Retrieved December 4, 2014, from http://www.cmhc-schl.gc.ca/odpub/esub/64291/64291_2012_B01.pdf?fr=1345862116234
- Celik, T., & Yetgin, Z. (2011). Change detection without difference image computation based on multiobjective cost function optimization. *Turk. J. of Elec. Eng. & Comp. Sci*, 19(6), 941-956.
- Chen, G., Hay, G. J., Carvalho, L. M., & Wulder, M. A. (2012). Object-based change detection. *International Journal of Remote Sensing*, 33(14), 4434-4457.
- Chen, J., Pappas, T. N., Mojsilovic, A., & Rogowitz, B. (2005). Adaptive perceptual color-texture image segmentation. *Image Processing, IEEE Transactions on*, 14(10), 1524-1536.
- Congalton, R. G. (1991). A review of assessing the accuracy of classifications of remotely sensed data. *Remote sensing of environment*, 37(1), 35-46.
- Coppin, P., Jonckheere, I., Nackaerts, K., Muys, B., & Lambin, E. (2004). Review Article Digital change detection methods in ecosystem monitoring: a review. *International journal of remote sensing*, 25(9), 1565-1596.
- Deng, J. S., Wang, K., Hong, Y., & Qi, J. G. (2009). Spatio-temporal dynamics and evolution of land use change and landscape pattern in response to rapid urbanization. *Landscape and Urban Planning*, 92(3), 187-198.
- DMTI, 2006, 2009 , 2012. Montreal Satellite StreetView, 3_1-9_7_MONTREAL-S3XM, Markham ON: DMTI Spatial Inc., 2006, 2009 and 2012.
- Egenhofer, M. J., & Al-Taha, K. K. (1992). Reasoning about gradual changes of topological relationships. In *Theories and methods of spatio-temporal reasoning in geographic space* (pp. 196-219). Springer Berlin Heidelberg.
- Erwig, M., Gu, R. H., Schneider, M., & Vazirgiannis, M. (1999). Spatio-temporal data types: An approach to modeling and querying moving objects in databases. *GeoInformatica*, 3(3), 269-296.
- Esbensen, K. H., Geladi, P. L., & Grahn, H. F. (1992). Strategies for multivariate image regression. *Chemometrics and intelligent laboratory systems*, 14(1), 357-374.

- Gallo, G., Grigoriadis, M. D., & Tarjan, R. E. (1989). A fast parametric maximum flow algorithm and applications. *SIAM Journal on Computing*, 18(1), 30-55.
- Gao, S., Li, L., Li, W., Janowicz, K., & Zhang, Y. (2014). Constructing gazetteers from volunteered big geo-data based on Hadoop. *Computers, Environment and Urban Systems*, doi:10.1016, in press.
- Hall, P., & Koch, I. (1992). On the feasibility of cross-validation in image analysis. *SIAM Journal on Applied Mathematics*, 52(1), 292-313.
- Hampton, S. E., Strasser, C. A., Tewksbury, J. J., Gram, W. K., Budden, A. E., Batcheller, A. L., ... & Porter, J. H. (2013). Big data and the future of ecology. *Frontiers in Ecology and the Environment*, 11(3), 156-162.
- Huang, B., & Chandramouli, M. (2009). Spatio-Temporal Object Modeling. *Handbook of Research on Geoinformatics*.
- Jermyn, I. H., & Ishikawa, H. (2001). Globally optimal regions and boundaries as minimum ratio weight cycles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(10), 1075-1088.
- Jenerette, G. D., & Wu, J. (2001). Analysis and simulation of land-use change in the central Arizona–Phoenix region, USA. *Landscape ecology*, 16(7), 611-626.
- Jianya, G., Haigang, S., Guorui, M., & Qiming, Z. (2008). A review of multi-temporal remote sensing data change detection algorithms. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37(B7), 757-762.
- Kolmogorov, V., Boykov, Y., & Rother, C. (2007). Applications of parametric maxflow in computer vision. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on* (pp. 1-8). IEEE.
- Lam, N. S. N. (1983). Spatial interpolation methods: a review. *The American Cartographer*, 10(2), 129-150.
- Langran, G., & Chrisman, N. R. (1988). A framework for temporal geographic information. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 25(3), 1-14.
- Ligmann - Zielinska, A., Church, R. L., & Jankowski, P. (2008). Spatial optimization as a generative technique for sustainable multiobjective land - use allocation. *International Journal of Geographical Information Science*, 22(6), 601-622.

- Lee, K. H., Lee, Y. J., Choi, H., Chung, Y. D., & Moon, B. (2012). Parallel data processing with MapReduce: a survey. *AcM SIGMOD Record*, 40(4), 11-20.
- Lempitsky, V., Blake, A., & Rother, C. (2012). Branch-and-mincut: global optimization for image segmentation with high-level priors. *Journal of Mathematical Imaging and Vision*, 44(3), 315-329.
- Li, H., Manjunath, B. S., & Mitra, S. K. (1995). A contour-based approach to multisensor image registration. *Image Processing, IEEE Transactions on*, 4(3), 320-334.
- Li, J., Bioucas-Dias, J. M., & Plaza, A. (2010). Semisupervised hyperspectral image segmentation using multinomial logistic regression with active learning. *Geoscience and Remote Sensing, IEEE Transactions on*, 48(11), 4085-4098.
- Li, Q., Zhang, T., & Yu, Y. (2011). Using cloud computing to process intensive floating car data for urban traffic surveillance. *International Journal of Geographical Information Science*, 25(8), 1303-1322.
- Li, X., & Yeh, A. G. O. (2002). Neural-network-based cellular automata for simulating multiple land use changes using GIS. *International Journal of Geographical Information Science*, 16(4), 323-343.
- Liang, S., Chen, S., Huang, C., Li, R., Chang, Y., Badger, J., & Rezel, R. (2010, September). Geocens: Geospatial cyberinfrastructure for environmental sensing. In *Proceedings of GIScience 2010—Sixth international conference on Geographic Information Science* (Vol. 6292). Zurich, Switzerland: Springer.
- Lillesand, T. M., Kiefer, R. W., & Chipman, J. W. (2004). *Remote sensing and image interpretation* (No. Ed. 5). John Wiley & Sons Ltd.
- Liu, X., Veksler, O., & Samarabandu, J. (2008). Graph cut with ordering constraints on labels and its applications. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (pp. 1-8). IEEE.
- Liu, Y., Hu, K., Zhu, Y., & Chen, H. (2014). A Novel Method for Image Segmentation Based on Nature Inspired Algorithm. In *Intelligent Computing in Bioinformatics* (pp. 390-402). Springer International Publishing.
- Lu, D., Mausel, P., Brondizio, E., & Moran, E. (2004). Change detection techniques. *International journal of remote sensing*, 25(12), 2365-2401.
- Madden, S., & Franklin, M. J. (2002). Fjording the stream: An architecture for queries over streaming sensor data. In *Data Engineering, 2002. Proceedings. 18th International Conference on Data Engineering* (pp. 555-566). IEEE.

- Malik, J., Belongie, S., Leung, T., & Shi, J. (2001). Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, 43(1), 7-27.
- McAfee A., Brynjolfsson E., "Big data: The management revolution. *Harvard Business Rev*, 90(10), 61-67.
- Macleod, R. D., & Congalton, R. G. (1998). A quantitative comparison of change-detection algorithms for monitoring eelgrass from remotely sensed data. *Photogrammetric Engineering and Remote Sensing*, 64(3), 207-216.
- Melgani, F., & Bruzzone, L. (2004). Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transactions on Geoscience and Remote Sensing* 42(8), 1778-1790.
- Miller, H. J., & Goodchild, M. F. (2014). Data-driven geography. *GeoJournal*, 1-13.
- Myint, S. W., Gober, P., Brazel, A., Grossman-Clarke, S., & Weng, Q. (2011). Per-pixel vs. object-based classification of urban land cover extraction using high spatial resolution imagery. *Remote Sensing of Environment*, 115(5), 1145-1161.
- Nadi, S., & Delavar, M. R. (2003). Spatio-Temporal Modeling of Dynamic Phenomena in GIS. In *ScanGIS* (pp. 215-225).
- Nandal, R. (2013). Spatio-Temporal Database and Its Models: A Review. *Journal of Computer Engineering (IOSR-JCE)*, 2278-0661, p-ISSN: 2278-8727, Volume 11, Issue 2 (May. -Jun. 2013), PP 91-100.
- Nurain, N., Sarwar, H., Sajjad, M. P., & Mostakim, M. (2012). An In-depth Study of Map Reduce in Cloud Environment. In *Advanced Computer Science Applications and Technologies (ACSAT)*, 2012 International Conference on (pp. 263-268). IEEE.
- Pan, D., Domon, G., De Blois, S., & Bouchard, A. (1999). Temporal (1958–1993) and spatial patterns of land use changes in Haut-Saint-Laurent (Quebec, Canada) and their relation to landscape physical attributes. *Landscape Ecology*, 14(1), 35-52.
- Parent, C., Spaccapietra, S., & Zimányi, E. (1999). Spatio-temporal conceptual models: data structures+ space+ time. In *Proceedings of the 7th ACM international symposium on Advances in geographic information systems* (pp. 26-33). ACM.
- Peuquet, D. J., & Duan, N. (1995). An event-based spatiotemporal data model (ESTDM) for temporal analysis of geographical data. *International journal of geographical information systems*, 9(1), 7-24.

- Pijanowski, B. C., Brown, D. G., Shellito, B. A., & Manik, G. A. (2002). Using neural networks and GIS to forecast land use changes: a land transformation model. *Computers, environment and urban systems*, 26(6), 553-575.
- Quesada, I., & Grossmann, I. E. (1992). An LP/NLP based branch and bound algorithm for convex MINLP optimization problems. *Computers & Chemical Engineering*, 16(10), 937-947.
- Radke, R. J., Andra, S., Al-Kofahi, O., & Roysam, B. (2005). Image change detection algorithms: a systematic survey. *Image Processing, IEEE Transactions on*, 14(3), 294-307.
- Ray, S., & Turi, R. H. (1999). Determination of number of clusters in k-means clustering and application in colour image segmentation. In *Proceedings of the 4th international conference on advances in pattern recognition and digital techniques* (pp. 137-143).
- Saalfeld, A. (1999). Topologically consistent line simplification with the Douglas-Peucker algorithm. *Cartography and Geographic Information Science*, 26(1), 7-18.
- Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8), 888-905.
- Singh, A. (1989). Review article digital change detection techniques using remotely-sensed data. *International journal of remote sensing*, 10(6), 989-1003.
- Sumengen, B., & Manjunath, B. S. (2006). Graph partitioning active contours (GPAC) for image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(4), 509-521.
- Tong, D., & Murray, A. T. (2012). Spatial optimization in geography. *Annals of the Association of American Geographers*, 102(6), 1290-1309.
- Van Der Wal, D., & Pye, K. (2003). The use of historical bathymetric charts in a GIS to assess morphological change in estuaries. *The Geographical Journal*, 169(1), 21-31.
- Veldkamp, A., & Lambin, E. F. (2001). Predicting land-use change. *Agriculture, ecosystems & environment*, 85(1), 1-6.
- Vincent, L., & Soille, P. (1991). Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE transactions on pattern analysis and machine intelligence*, 13(6), 583-598.

- Walter, V. (2004). Object-based classification of remote sensing data for change detection. *ISPRS Journal of photogrammetry and remote sensing*, 58(3), 225-238.
- Wang, M. (2014). An Improved Image Segmentation Algorithm Based on Principal Component Analysis. In *Proceedings of the 9th International Symposium on Linear Drives for Industry Applications*, Volume 4 (pp. 811-819). Springer Berlin Heidelberg.
- Wang, S. (2010). A CyberGIS framework for the synthesis of cyberinfrastructure, GIS, and spatial analysis. *Annals of the Association of American Geographers*, 100(3), 535-557.
- Winn, J., & Shotton, J. (2006). The layout consistent random field for recognizing and segmenting partially occluded objects. In *Computer Vision and Pattern Recognition*, 2006 IEEE Computer Society Conference on (Vol. 1, pp. 37-44). IEEE.
- Wu, Z., & Leahy, R. (1993). An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(11), 1101-1113.
- Xia, Y. J., Kuang, L., & Li, X. M. (2011). Accelerating geospatial analysis on GPUs using CUDA. *Journal of Zhejiang University SCIENCE C*, 12(12), 990-999.
- Xing, J., Sieber, R., & Kalacska, M. (2014). The challenges of image segmentation in big remotely sensed imagery data. *Annals of GIS*, 20(4), 233-244.
- Xing, J., & Sieber, R. (2014). Sampling Based Image Splitting In Large Scale Distributed Computing Of Earth Observation Data, 2014. *Proceedings 35th IEEE Geoscience and Remote Sensing Symposium*. IEEE.
- Yang, C., Goodchild, M., Huang, Q., Nebert, D., Raskin, R., Xu, Y., ... & Fay, D. (2011). Spatial cloud computing: how can the geospatial sciences use and help shape cloud computing?. *International Journal of Digital Earth*, 4(4), 305-329.
- Yang, C., Raskin, R., Goodchild, M., & Gahegan, M. (2010). Geospatial cyberinfrastructure: past, present and future. *Computers, Environment and Urban Systems*, 34(4), 264-277.
- Yang, Z., Zhu, Y., & Pu, Y. (2008). Parallel image processing based on CUDA. In *Computer Science and Software Engineering*, 2008 International Conference on (Vol. 3, pp. 198-201). IEEE.

- Yuan, M. (1996, January). Temporal GIS and spatio-temporal modeling. In Proceedings of Third International Conference Workshop on Integrating GIS and Environment Modeling, Santa Fe, NM.
- Yuan, M. (1999). Use of a Three Domain Representation to Enhance GIS Support for Complex Spatiotemporal Queries. *Transactions in GIS*, 3(2), 137-159.
- Yue, P., Gong, J., & Di, L. (2010). Augmenting geospatial data provenance through metadata tracking in geospatial service chaining. *Computers & Geosciences*, 36(3), 270-281.
- Zaslavsky, A., Perera, C., & Georgakopoulos, D. (2013). Sensing as a service and big data. arXiv preprint arXiv:1301.0159.

Figures and Images

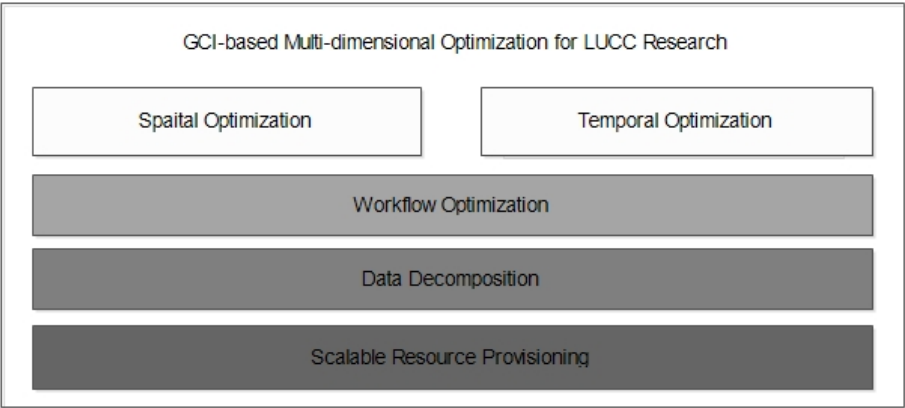


Figure 1. GCI-based Multi-dimensional Optimization for LUCC Research

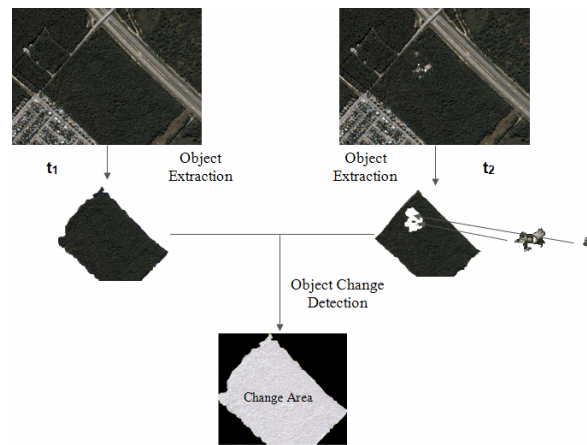


Figure 2. Example of drawbacks of OBIA Change Detection. Fine grained changes occurring at t_2 will fail to be recorded when compared to change area at t_1 . Samples are extracted from Montreal Streetview satellite images 2006 and 2009 (Source: DMTI Inc.).

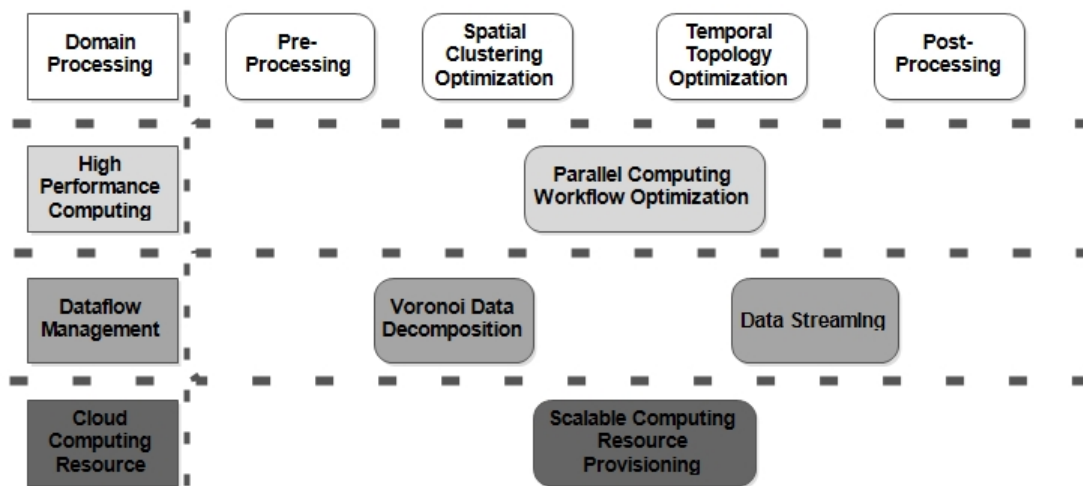


Figure 3. Lucc-based GCI and the Multi-dimensional Optimization

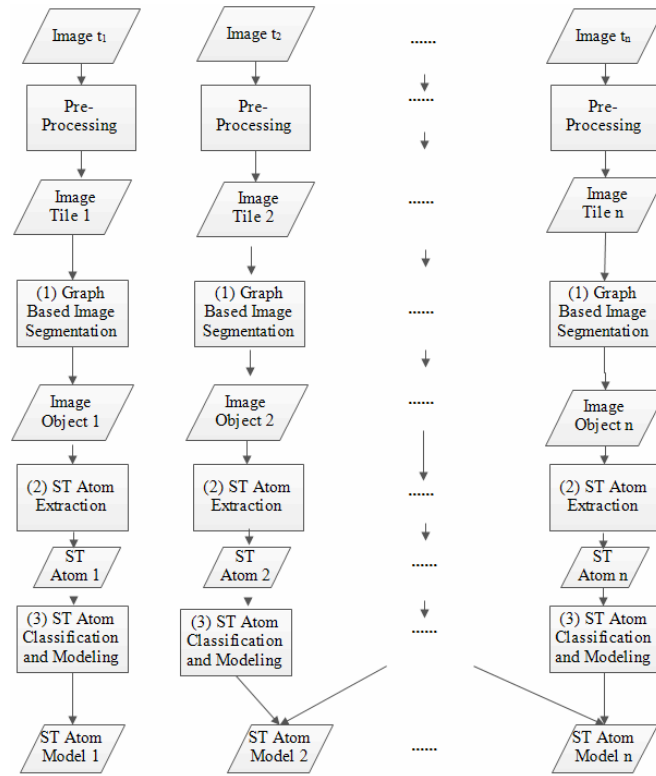


Figure 4. Workflow of our LUCC Framework, with input/output illustration of each step.

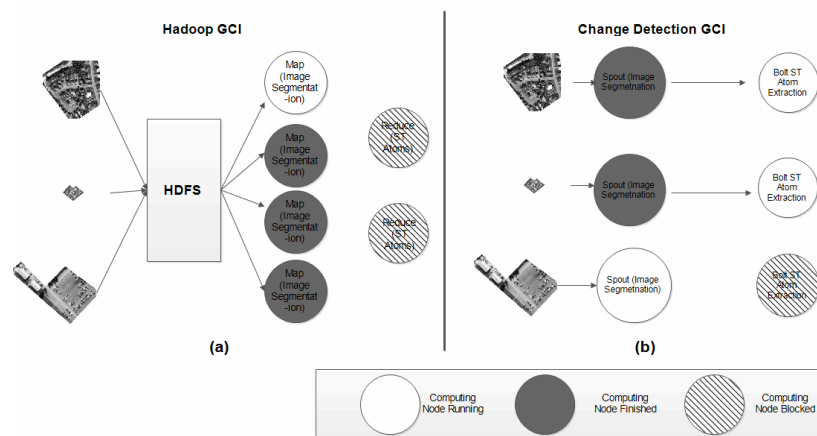


Figure 5. Comparison Between Hadoop GCI and Apache Storm in Change Detection GCI

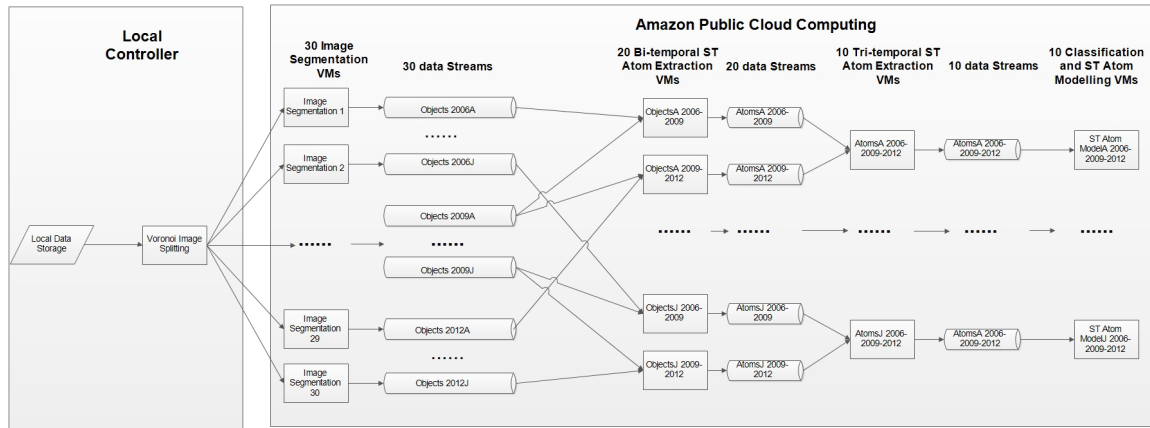


Figure 6. Implementation Details of Change Detection GCI for Montreal 2006-2012. The test bed includes 70 VMs in Amazon EC2 and 60 Amazon Kinesis data streams.

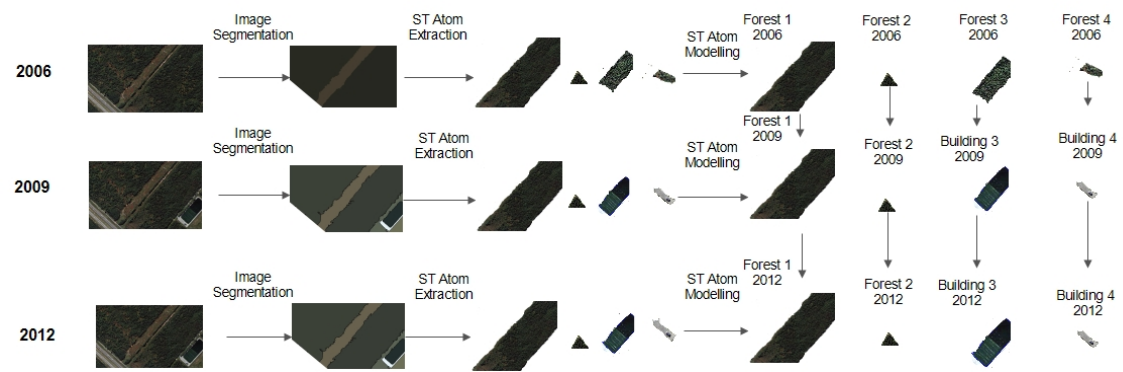


Figure 7. Examples. of forest changing to buildings in suburban area with results of image segmentation, ST atom extraction, and ST atom classification and modelling. Streetview image collected in 2006, 2009, and 2012 from Greater Montreal Area (Source: DMTI Inc.)

Table 1. Temporal Topology Rules Chosen for ST Atoms Extraction




















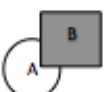


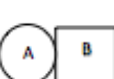


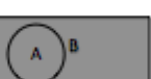

Temporal Topology Relationship	Study Time t_1	Change Reference Time t_2	Explanation	ST Object Atom
Equal			The object A has changed completely to another object in t_2 .	
Split			Object A has changed to two or more objects in t_2 .	
Partial Change			One or more parts of object A has changed to another objects in t_2 .	
Contain			One or more changes happen within object A in t_2 .	
Expand			Although object A become larger in t_2 , the ST object atom is till its original size.	
Shrink			Object A become smaller in t_2 .	
Overlap			Objects B changes its position and overlaps with A in t_2 .	
Merge			The adjacent object or parts have changed to A in t_2 .	
Covered-By			Object A is totally covered by its neighbor object B in t_2 .	

Table 2 Testbed Configurations

	Number of VMs	Instance Type	VCPU	VMemory	Local Storage
Pre-Processing and Image Decomposition	1	Private Cloud	8	32.0	1TB
Image Segmentation	30	m3.large	2	7.5	32 GB SSD
ST Atom Extraction	30	m3.large	2	7.5	32 GB SSD
Classification and ST Atom Modelling	10	m3.large	2	7.5	32 GB SSD

Table 3. Computing Time for Steps in Change Detection Implementation. Then compared to a second implementation in which Hadoop replaces Storm.

Steps	Change Detection GCI (Hours)	Hadoop based GCI (Hours)
Pre-Processing and Image Decomposition	91.4	91.4
Image Segmentation with Optimization	19.7	26.5
ST Atom Extraction	4.6	19.3
Classification and Modeling	14.4	31.5
Total	130.1	168.7

Table 4. Comparison between min-cut/max-flow and branch-and-mincut optimization algorithms

		2006-2009 Reference			2009-2012 Reference		
		Change	No-	Total	Change	No-	Total
		(%)	Change	(%)	(%)	Change	(%)
			(%)			(%)	
min-cut/max-flow	Change	6.7	1.4	8.1	5.2	0.8	6.0
	(%)						
	No-Change	1.4	90.5	91.9	2.6	91.4	94.0
	(%)						
	Total (%)	8.7	91.3	100.0	7.8	92.2	100.0
branch-and-mincut	Change	6.9	1.2	8.1	5.3	0.7	6.0
	(%)						
	No-Change	0.8	91.1	91.9	2.0	92.0	94.0
	(%)						
	Total (%)	7.7	92.3	100.0	7.5	92.7	100.0